Fast, bias-free tracking of single particles with variable size and shape

Andrew J. Berglund, Matthew D. McMahon, Jabez J. McClelland, and J. Alexander Liddle

Center for Nanoscale Science and Technology, National Institute of Standards and Technology, Gaithersburg, MD 20899

andrew.berglund@nist.gov

Abstract: We introduce a fast and robust technique for single-particle tracking with nanometer accuracy. We extract the center-of-mass of an image with a simple, iterative algorithm that efficiently suppresses background-induced bias in a simplistic centroid estimator. Unlike many commonly used algorithms, our position estimator requires no prior information about an image and uses only simple arithmetic operations, making it appropriate for future hardware implementation and real-time feedback applications. We demonstrate it both numerically and experimentally, using an inexpensive CCD camera to localize 190 nm fluorescent microspheres to better than 5 nm.

OCIS codes: (180.2520) Fluorescence microscopy, (100.2960) Image analysis

References and links

1. M. J. Saxton and K. Jacobson, "Single-particle tracking: applications to membrane dynamics," Annu. Rev. Biophys. Biomolec. Struct. **26**, 373–399 (1997).
2. A. Yildiz, J. N. Forkey, S. A. McKinney, T. Ha, Y. E. Goodman, and P. R. Selvin, "Myosin V walks hand-over-hand: Single fluorophore imaging with 1.5-nm localization," Science **300**, 2061–2065 (2003).
3. X. Michalet, F. F. Pinaud, L. A. Bentolila, J. M. Tsay, S. Doose, J. J. Li, G. Sundaresan, A. M. Wu, S. S. Gambhir, and S. Weiss, "Quantum Dots for Live Cells, in Vivo Imaging, and Diagnostics," Science **307**(5709), 538–544 (2005).
4. D. Weihs, T. G. Mason, and M. A. Teitell, "Bio-Microrheology: A Frontier in Microrheology," Biophys. J. **91**(11), 4296–4305 (2006).
5. P. Bahukudumbi and M. A. Bevan, "Imaging energy landscapes with concentrated diffusing colloidal probes," The Journal of Chemical Physics **126**(24), 244702 (pages 7) (2007). URL http://link.aip.org/link/?JCP/126/244702/1.
6. H.-J. Wu, W. Everett, S. Anekal, and M. Bevan, "Mapping Patterned Potential Energy Landscapes with Diffusing Colloidal Probes," Langmuir **22**(16), 6826–6836 (2006).
7. S. K. Sainis, V. Germain, and E. R. Dufresne, "Statistics of Particle Trajectories at Short Time Intervals Reveal fN-Scale Colloidal Forces," Physical Review Letters **99**(1), 018303 (pages 4) (2007). URL http://link.aps.org/abstract/PRL/v99/e018303.
8. N. Bobroff, "Position measurement with a resolution and noise-limited instrument," Rev. Sci. Instr. **57**, 1152 (1986).
9. M. Cheezum, W. Walker, and W. Guilford, "Quantitative Comparison of Algorithms for Tracking Single Fluorescent Particles," Biophys. J. **81**(4), 2378–2388 (2001).
10. R. Thompson, D. Larson, and W. Webb, "Precise Nanometer Localization Analysis for Individual Fluorescent Probes," Biophys. J. **82**(5), 2775–2783 (2002).
11. J. Crocker and D. Grier, "Methods of Digital Video Microscopy for Colloidal Studies," J. Colloid Interface Sci. **179**(1), 298–310 (1996).
12. V. Levi, Q. Ruan, and E. Gratton, "3-D particle tracking in a two-photon microscope. Application to the study of molecular dynamics in cells," Biophys. J. **88**, 2919–2928 (2005).
13. A. J. Berglund and H. Mabuchi, "Tracking-FCS: Fluorescence Correlation Spectroscopy of Individual Particles," Opt. Express **13**, 8069–8082 (2005).

14. M. Armani, S. Chaudhary, R. Probst, and B. Shapiro, “Using feedback control and micro-fluidics to steer individual particles,” Micro Electro Mechanical Systems, 2005. MEMS 2005. 18th IEEE International Conference on pp. 855–858.
15. A. E. Cohen and W. E. Moerner, “Method for Trapping and Manipulating Nanoscale Objects in Solution,” Appl. Phys. Lett. **86**(9), 093,109 (2005).
16. H. Cang, C. M. Wong, C. S. Xu, A. H. Rizvi, and H. Yang, “Confocal three dimensional tracking of a single nanoparticle with concurrent spectroscopic readout,” Appl. Phys. Lett. **88**, 223,901 (2006).
17. K. McHale, A. J. Berglund, and H. Mabuchi, “Quantum dot photon statistics measured by three-dimensional particle tracking,” Nano Letters **7**, 3535–3539 (2007).
18. M. Armani, S. Chaudhary, R. Probst, and B. Shapiro, “Using feedback control of microflows to independently steer multiple particles,” IEEE J. Microelectromech. Syst. **15**(4), 945–956 (2006).
19. L. Novotny and B. Hecht, *Principles of Nano-Optics* (Cambridge University Press, 2006).

1. Introduction

In single-particle tracking experiments, fluorescent molecules, quantum dots, metal nanoparticles, and polymer microspheres are routinely localized on a 10 nm length scale, and in exceptional cases even on a 1 nm length scale, despite the much longer optical wavelength encoding that information. This ability to track the nanometer-scale motion of small objects with optical microscopy has greatly improved our experimental access to nanoscale biological behavior [1, 2, 3, 4] and microscale-to-nanoscale colloid interactions [5, 6, 7].

In most cases, particle tracking is accomplished by analyzing a series of digital images, for example the output from a CCD camera. However, the analysis of these images presents a difficult data processing task, and a correspondingly large body of research has been devoted to developing and testing particle-tracking software routines. When selecting an algorithm for extracting the underlying position of a particle from a noisy image, researchers have been forced to accept tradeoffs among accuracy, robustness, and speed. For example, it is generally accepted that the most accurate results can be achieved by fitting an image to a constrained (usually Gaussian) instrument resolution function [8, 9, 10]. Unfortunately, the computational complexity of such an algorithm renders it quite slow and therefore unavailable for real-time or time-critical applications. Other algorithms offer much shorter execution times by incorporating prior knowledge of the image shape through masking [10] or spatial filtering [11]. However, these methods are not robust to variations in the size or shape of the tracked object that might arise when tracking an asymmetric object such as a nanorod or when a single particle diffuses into and out of focus.

Moreover, recently developed experimental techniques for manipulating nanoparticles using feedback control [12, 13, 14, 15, 16, 17] place stringent requirements on the speed and accuracy of particle-tracking routines. Because image processing is simply too slow, many particle-tracking feedback controllers use photon counting signals rather than CCD imaging methods to localize particles in real time. However, real-time control of multiple, spatially-separated particles [18] demands an imaging method and will be directly enhanced by the development of simple, fast, and accurate algorithms for localizing individual particles within subsections of a larger image. Future experiments that combine nanometer-resolution particle localization with real-time feedback control will enable new techniques for monitoring and engineering structures at the nanoscale.

In this paper, we progress toward this goal of robust, real-time particle tracking by introducing a simple, fast estimator for extracting the center position of an object from an image corrupted by noise, pixelation, and a constant (unknown) background. Our algorithm is a refinement of the simple centroid or center-of-mass (CM) algorithm, which is very fast to execute but only provides an unbiased estimate when an object happens to lie exactly at the center of the image. In the estimator algorithm developed here, we enforce this otherwise serendipitous arrangement by iteratively testing whether an object lies at the image center and subsequently refining the image window in order to better center the object. This procedure efficiently suppresses the estimator bias (exponentially in the number of iterations), resulting in a very fast, unbiased localization algorithm with sub-pixel accuracy. It requires *no input parameters* and makes *no assumptions* about the object shape, making it robust and effective for localizing objects with varying sizes or complex shapes. Furthermore, the algorithm is computationally simple and can be executed with a few lines of code. Its performance is comparable to or exceeds full nonlinear least-squares minimization in many cases, while its execution time is orders of magnitude shorter. For all of these reasons, our algorithm is a promising candidate for implementation in signal-processing hardware for real-time applications.

The paper is organized as follows. In section 2, we examine the error in the CM estimator in detail and show that this error can be estimated in real-time. In section 3, we define an iterative "Virtual Window" CM (VWCM) algorithm that exploits this fact to eliminate the average error in the CM estimator. In section 4, we present the results of extensive Monte Carlo simulations of the new estimator, and in section 5 we apply the VWCM in our own particle-tracking experiment. Finally, section 6 summarizes our results and provides an outlook for future applications and refinements.

2. Bias in the Center-of-Mass Estimator

As mentioned in section 1, the center-of-mass (CM) estimator of an object's position has many desirable characteristics but significant drawbacks as well. It is simple, very fast to execute, and provides an estimate with no assumptions about the shape or size of the underlying object. Unfortunately, it is strongly biased by any background signal and exhibits poor noise rejection outside the region of interest. In this section, we will derive the mean error, or bias, in the CM estimator and show that this error can itself be estimated in real-time.

With the output of a CCD camera in mind, we define an image to be a two-dimensional matrix of intensities $S = S_{jk}$ representing the total number of counts in pixel P_{jk} with width Δ centered at location $(x,y) = (x_{jk}, y_{jk})$.[1] We assume the underlying signal is drawn from a Poisson distribution with spatially-dependent density $N_S(x,y)/\Delta^2$, which represents the point-spread function of the optical system convolved with the shape of the fluorescent object in units of counts per pixel area. For a general scenario, we can also include a background signal with mean $N_B(x,y)/\Delta^2$ and (spatially-uncorrelated) variance $\sigma_B^2(x,y)/\Delta^2$. The noise term may represent scattered light or technical noise in the camera, with statistical properties (e.g. Poisson or Gaussian distribution) absorbed into the definitions of N_B and σ_B^2. Mathematically, we can now write the mean and covariance of the signal from a particle at position (x_0, y_0) as integrals over each pixel:

$$\begin{aligned} \langle S_{jk} \rangle &= \iint_{P_{jk}} \frac{dxdy}{\Delta^2} \left[N_S(x - x_0, y - y_0) + N_B(x,y) \right] && (1) \\ \langle S_{jk} S_{j'k'} \rangle &= \langle S_{jk} \rangle \langle S_{j'k'} \rangle + \delta_{jj'} \delta_{kk'} \iint_{P_{jk}} \frac{dxdy}{\Delta^2} \left[N_S(x - x_0, y - y_0) + \sigma_B^2(x,y) \right]. && (2) \end{aligned}$$

Now consider the CM estimator of the object's x-position x_0:

$$\hat{x}_{CM} = \frac{\sum_{jk} x_{jk} S_{jk}}{\sum_{jk} S_{jk}}. \tag{3}$$

[1] The pixel size Δ and coordinates (x,y) are always considered in the object plane of the optical system, so that, for example, Δ is the actual CCD pixel size divided by the system magnification M.

The normalization factor (denominator) makes $\hat{x}_{CM}$ a nonlinear function of the image S_{jk}, however a linearized approximation is accurate to order $\mathscr{N}^{-1/2}$, where $\mathscr{N} = \sum_{jk} S_{jk}$ is the *total* number of counts in the image. The linearized approximation to Eq. (3) is

$$\hat{x}_{CM} = \frac{1}{\langle \mathscr{N} \rangle} \sum_{jk} x_{jk} S_{jk} \quad , \quad \langle \mathscr{N} \rangle = \sum_{jk} \langle S_{jk} \rangle \,. \tag{4}$$

We can now calculate the mean and variance of the centroid estimator for an image S_{jk} satisfying (1):

$$\langle \hat{x}_{CM} \rangle \;=\; \frac{1}{\langle \mathscr{N} \rangle} \sum_{jk} \left\{ x_{jk} \iint_{P_{jk}} \frac{dxdy}{\Delta^2} \left[N_S(x - x_0, y - y_0) + N_B(x,y) \right] \right\} \tag{5}$$

$$\langle \hat{x}^2_{CM} \rangle - \langle \hat{x}_{CM} \rangle^2 \;=\; \frac{1}{\langle \mathscr{N} \rangle^2} \sum_{jk} \left\{ x^2_{jk} \iint_{P_{jk}} \frac{dxdy}{\Delta^2} \left[N_S(x - x_0, y - y_0) + \sigma^2_B(x,y) \right] \right\} . \tag{6}$$

Eqs. (5)-(6) allow us to calculate the statistics of the centroid estimator for any underlying image function $N_S(x,y)$ superimposed on a spatially-varying, Poisson- or Gaussian-distributed background. The mean and mean-square errors (bias and variance) resulting from pixelation, truncation, shot noise, and background noise can each be derived from these expressions.

Let us now calculate the bias arising from a spatially-invariant, though noisy, background signal with $N_B(x,y) = N_B$ and $\sigma^2_B(x.y) = \sigma^2_B$. We will assume that the particle is near the center of the image and is not significantly truncated at the edges of the array. Introducing the shorthand notation,

$$N^\Delta_S(x_{jk} - x_0, y_{jk} - y_0) = \iint_{P_{jk}} \frac{dxdy}{\Delta^2} N_S(x - x_0, y - y_0)$$

and rearranging Eq. (5), we find that the average CM estimator error, the bias B_{CM}, can be written as

$$\begin{aligned} B_{CM} &\equiv \langle \hat{x}_{CM} - x_0 \rangle \\ &= \frac{1}{\sum_{jk} N^\Delta_S(x_{jk} - x_0, y_{jk} - y_0)} \left[\sum_{jk} (x_{jk} - x_0) N^\Delta_S(x_{jk} - x_0, y_{jk} - y_0) + N_B \sum_{jk} (x_{jk} - \langle \hat{x}_{CM} \rangle) \right] . \end{aligned} \tag{7}$$

Let us now consider the different contributions to the bias in Eq. (7). The normalization factor $\sum_{jk} N^\Delta_S(x_{jk} - x_0, y_{jk} - y_0)$ represents the average number of detected counts from the particle itself, and is only a weak function of x_0 and y_0 as long as the particle image is not strongly truncated at the edges of the array. We may then denote this term by $\langle \mathscr{N}_S \rangle$ and neglect its functional dependence on (x_0, y_0). The first term in brackets on the right-hand side of Eq. (7) is a discretized analog of the continuous center of mass

$$\sum_{jk} (x_{jk} - x_0) N^\Delta_S(x_{jk} - x_0, y_{jk} - y_0) \approx \iint \frac{dxdy}{\Delta^2} x N_S(x,y) = 0. \tag{8}$$

The last integral is proportional to the x-coordinate of the center of mass of the image function $N_S(x,y)$, which we assume to be 0 [this condition can always be enforced through the definition of $N_S(x,y)$]. Violations of the approximate equality in Eq. (8) correspond to estimator bias arising from pixelation and truncation of the underlying image. Assuming these are negligible for now, we are left with the following expression for the estimator bias

$$B_{CM} \approx \frac{\langle \mathscr{N}_B \rangle}{\langle \mathscr{N}_S \rangle} \left(\bar{x} - \langle \hat{x}_{CM} \rangle \right) \tag{9}$$

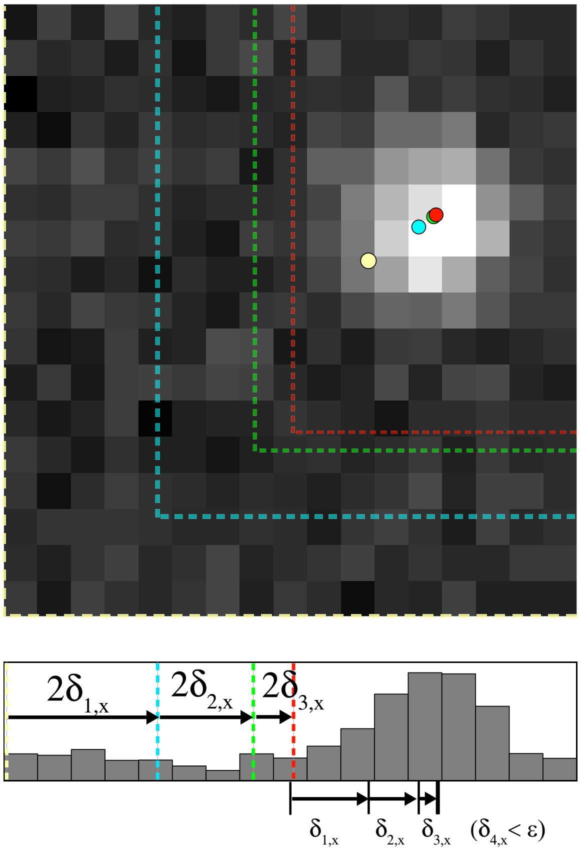

Fig. 1. Cartoon illustration of the iterative VWCM algorithm operating on an image of a particle with a constant background. The background biases the center-of-mass (CM) estimate towards the center of the array. The first, biased centroid estimate (yellow) is offset by $\delta_{1,x}$ from the array center. At the second iteration, the window is truncated by an amount $2\delta_{1,x}$ along x and a new centroid (blue) is calculated within this window. Where part of a pixel is truncated by the "virtual window," its value is scaled proportionally to the relative area. At each iteration, the window is further adjusted until the center of the window and the CM estimate coincide, giving a bias-free estimate of the particle position.

where $\langle \mathcal{N}_S \rangle$ and $\langle \mathcal{N}_B \rangle$ are the average number of signal and background counts in the entire image, respectively, and $\bar{x}$ is the geometric center of the pixel array, *i.e.* the unweighted average x-coordinate. Eq. (9) reveals the key fact that *the estimator bias is proportional to the difference between the center of the pixel array* $\bar{x}$ *and the average estimate* $\langle \hat{x}_{CM} \rangle$. The constant of proportionality is the ratio of background to signal counts in the image. Recall that the estimator bias is unknown to the experimenter, since the underlying particle position x_0 is unknown. However, the difference between the estimate $\hat{x}_{CM}$ and the center of the pixel array can be calculated in each shot of the experiment. In our algorithm described below, we exploit this fact in order to form an online estimate of the bias then correct it by truncating the pixel window in order to center the particle in the image array.

3. Virtual Window Centroid Algorithm

In section 2, we found that the bias in the CM estimator $\hat{x}_{CM}$ can itself be estimated in real time. The central concept of our Virtual Window Center-of-Mass (VWCM) estimator is to use this information to modify the image window in order to center it on the object and eliminate the estimator bias. The procedure is iterative, such that at the nth iteration, we calculate the center of mass $\hat{x}^{(n)}_{CM}$ then modify the image array by eliminating a portion of the image near one edge, effectively shifting the geometric center $\bar{x}^{(n)}$ of the window. The update rule that defines our iterative algorithm is to truncate the window at the nth iteration such that $\bar{x}^{(n+1)} = \hat{x}^{(n)}_{CM}$.

To see how eliminating a portion of the image shifts the array center, consider eliminating a single row of pixels (width Δ) from the edge of an array. The center of the resulting (rectangular) array is shifted by $\Delta/2$ as a result. If our image resolution were infinitely fine, we could translate the array center $\bar{x}$ any desired amount by discarding arbitrarily small portions of the image at one edge. Since we do not have infinitely fine resolution in practice, we may still approximate a sub-pixel truncation of the array by weighting the pixels and pixel coordinates along one edge of the image. For example, if we wish to translate the image array by a small amount $\delta/2 < \Delta/2$, we simply multiply the pixel intensities S_{jk} at the one edge by $1-\delta/\Delta$ and redefine the coordinates along that edge by $x_{jk} \to x_{jk} + \delta$. This procedure approximates the truncation of a region of width δ from the negative edge of the image, defining a virtual window shifted by $\delta/2$ as desired: $\bar{x} \to \bar{x} \pm \delta/2$. To complete the algorithm, the user sets two termination conditions, ε and n_{max}, such that the algorithm terminates when $|\bar{x}^{(n+1)} - \bar{x}^{(n)}|/\Delta < \varepsilon$ or the number of iterations exceeds n_{max}. With these concepts, we can now precisely describe the VWCM algorithm, displayed graphically in figure 1:

Virtual Window Center-of-Mass (VWCM) algorithm

1. Let the image $S^{(1)} = S$ and coordinates $(x^{(1)}_{jk}, y^{(1)}_{jk}) = (x_{jk}, y_{jk})$ correspond to the raw data.

2. Beginning with $n = 1$, Calculate the center-of-mass $\hat{x}^{(n)}_{CM}$ from the image $S^{(n)}$ and coordinates $x^{(n)}_{jk}$:

$$\hat{x}^{(n)}_{CM} = \frac{\sum_{jk} x^{(n)}_{jk} S^{(n)}_{jk}}{\sum_{jk} S^{(n)}_{jk}}.$$

   Repeat for $\hat{y}^{(n)}_{CM}$.

3. Define a new image $S^{(n+1)}$ and new coordinate system $(x^{(n+1)}_{jk}, y^{(n+1)}_{jk})$ by truncating the previous image such that

$$\bar{x}^{(n+1)} = \hat{x}^{(n)} \quad , \quad \bar{y}^{(n+1)} = \hat{y}^{(n)}.$$

   For subpixel shifts, use the virtual window procedure described above.

4. Iterate until $|\bar{x}^{(n+1)} - \bar{x}^{(n)}| < \varepsilon$ or $n = n_{max}$.

The update rule $\bar{x}^{(n+1)} = \hat{x}^{(n)}_{CM}$ corresponds to a shift or distortion of the image that centers the array on the current estimate of the particle position. Denoting the bias in the nth iteration by $B^{(n)}_{CM} = \langle \hat{x}^{(n)}_{CM} \rangle - x_0$, we find from Eq. (9) and the update rule $\bar{x}^{(n+1)}_{CM} = \hat{x}^{(n)}_{CM}$

$$B^{(n)}_{CM} = \left(\frac{1}{1 + \langle \mathcal{N}_S \rangle / \langle \mathcal{N}_B \rangle} \right)^{n-1} B^{(1)}_{CM} \tag{10}$$

where $\langle \mathcal{N}_S \rangle$ and $\langle \mathcal{N}_B \rangle$ are the average number of counts arising from the signal and background respectively. Eq. (10) shows that *the bias in the VWCM algorithm tends exponentially to zero with the number of iterations n*. The convergence rate depends on the signal-to-background ratio $\langle \mathcal{N}_S \rangle / \langle \mathcal{N}_B \rangle$, but the algorithm requires no knowledge of this quantity. In fact, the algorithm requires no input beyond the image S and pixel coordinates (x_{jk}, y_{jk}), and termination conditions ε and n_{max}. Finally, note that the algorithm requires only simple arithmetic operations on the image and is therefore very fast, particularly when the signal-to-background ratio gives a satisfactory convergence rate.

4. Numerical Simulations

We performed extensive numerical simulations in order to confirm the predicted features of the VWCM algorithm and compare its performance to other algorithms. For this test, we implemented four algorithms in standard numerical analysis software. We chose to compare the Gaussian fit, Gaussian mask [10], CM and VWCM. We included the Gaussian fit, because it is generally assumed to provide superior accuracy, and the Gaussian mask and CM because these are among the fastest executing algorithms and therefore most promising for real-time implementation.

For the Gaussian fit, we used a function of the form

$$f(x,y) = A \exp\left[-\frac{(x-x_0)^2}{2\sigma_x^2} - \frac{(y-y_0)^2}{2\sigma_y^2} \right] + B \tag{11}$$

where x_0, y_0, σ_x, σ_y, A, and B were the fit parameters. For both the Gaussian mask (see Ref. [10] for details) and the VWCM, we used termination conditions $\varepsilon = 10^{-3}$ and $n_{max} = 200$. Finally, we preprocessed every image array S passed to the algorithms by subtracting the minimum value from the entire array. This simple procedure ensures that the counts in each pixel are nonnegative (a prerequisite for the VWCM algorithm) and removes spuriously large offsets, which severely degrade the CM algorithm, make the Gaussian fit more sensitive to its initial fit conditions, and require more iterations to reach convergence in the VWCM.

A graphical summary of our simulations is shown in Fig. 2. In each case, we generated underlying count rate functions $N_S(x,y)$ corresponding to fluorescent objects with varying brightness, shape and position, as described in the figure caption. We then integrated $N_S(x,y)$ over the pixel coordinates to generate the pixelated rate $N_S^\Delta(x_{jk}, y_{jk})$. For each image realization, an array of Poisson-distributed random numbers was generated based on this rate. Finally, we added constant Poisson-distributed background noise, with a rate determined by the user-defined signal-to-background ratio $\langle \mathcal{N}_S \rangle / \langle \mathcal{N}_B \rangle$. The pixel size was taken to be $\Delta = 123$ nm, corresponding to our experiment (see section 5). For each type of object, we varied the position across a square grid corresponding to one pixel, generating a set of 1000 images at each position. The underlying object position was estimated for each image using the four algorithms, and each resulting distribution is plotted in Fig. 2 as a circle centered at the mean estimated position with a radius of one standard deviation (1σ).

Note that the Gaussian fit requires an initial parameter set that strongly affects its convergence, while the Gaussian mask requires a mask size and initial position guess in order to execute. For each particular image, the Gaussian mask's performance can be optimized by adjusting the mask size. Because robustness and automation are a key performance objective for our algorithm development, in Fig. 2 we optimized the Gaussian mask for the tightly focused image then used that mask shape for the remaining images without further "by-hand" adjustment. In contrast to the Gaussian fit and mask algorithms, the VWCM accepts no input other than the image matrix S and is reliable and easy to use in practice.

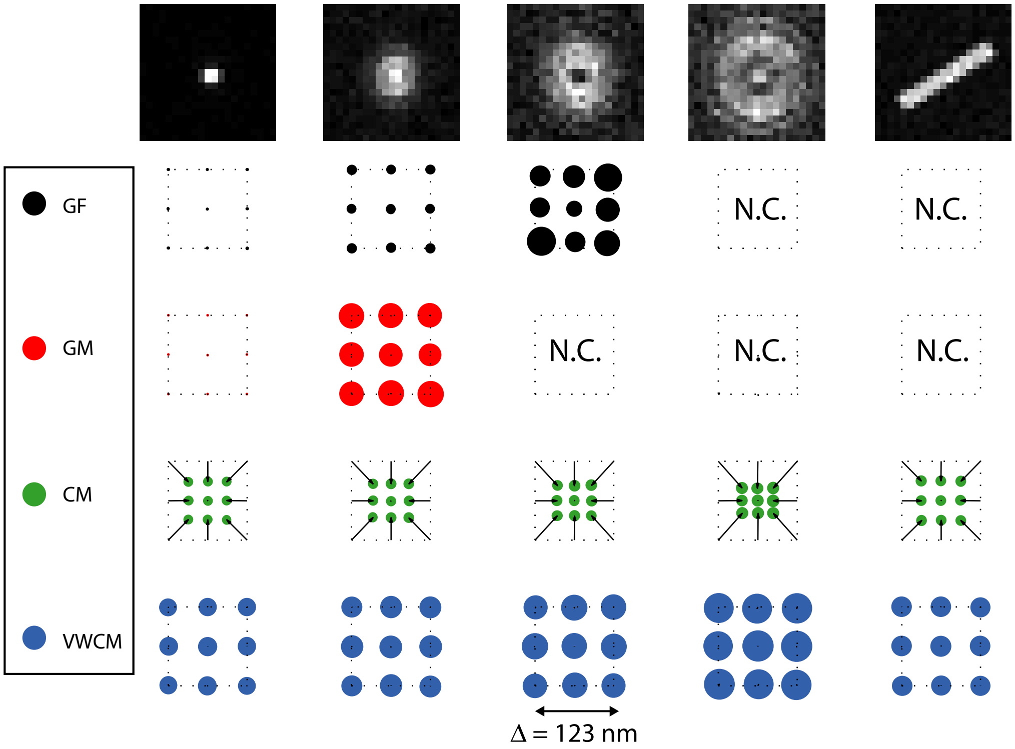

Fig. 2. Simulation results showing the bias and localization accuracy achieved for 21×21 pixel images using the Gaussian fit (GF), Gaussian mask (GM), CM and VWCM algorithms. For each image, the underlying position was varied over a grid of positions within the center pixel (dashed line). The resulting distribution of positions is displayed as a circle centered at the mean with 1σ radius as defined in the text. "N.C." denotes instances when the Gaussian fit and mask algorithms either failed to converge or had large errors with the resulting 1σ circles larger than the pixel. Image details: The first four images represent the (non-paraxial) point-spread function of a dipole emitter [19] with wavelength $\lambda = 550$ nm, at depths $z = 0$ nm, 500 nm, 750 nm and 1000 nm imaged through a microscope with magnification $M = 60$ and numerical aperture 1.2. The final image is a simple rod shape. Each image has $\langle \mathscr{N}_S \rangle = \langle \mathscr{N}_B \rangle = 5000$ photons.

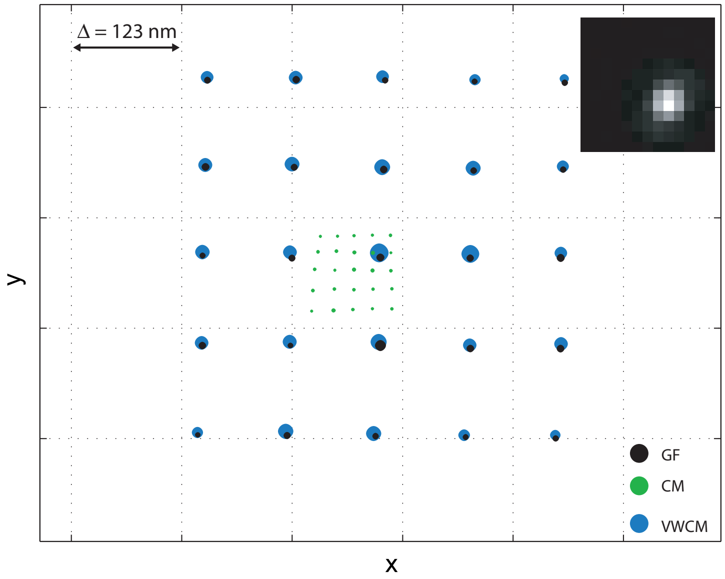

Fig. 3. Positions of a single 190 nm fluorescent microparticle stepped in 100 nm increments as estimated by the Gaussian fit (GF, black), CM (green), and VWCM (blue). Results are plotted as in Fig. 2, but with 2σ radii for better visibility. Note the extreme distortion introduced by bias in the CM algorithm. As a typical example, we quote the following results for the upper left position: the bias along x (measured from the Gaussian fit estimate) B_x, standard deviations along x (σ_x) and algorithm execution time (T). GF: $B_x = 0$ nm, $\sigma_x =$ 1.0 nm, $T = 66$ ms. CM: $B_x = 127$ nm, $\sigma_x = 0.6$ nm, $T = 0.2$ ms. VWCM: $B_x = -1.3$ nm, $\sigma_x = 2.2$ nm, T =2.7 ms.

Overall, the VWCM is fast, accurate, and robust. None of the other algorithms we investigated shared this combination of properties: the Gaussian fit algorithm has excellent overall accuracy and precision but is very slow and fails to converge in some cases; the Gaussian mask algorithm is both fast and accurate for a known, nearly-Gaussian image shape but its performance is significantly degraded when the mask is ill-suited to the object; the CM algorithm is the fastest, but exhibits a severe bias towards the geometric center of the image array. As displayed in figure 2, the VWCM completely eliminates the bias in the CM algorithm and performs robustly for a variety of underlying object shapes and sizes. It is less accurate than the Gaussian mask and Gaussian fit for tightly focused images within a large field but exhibits comparable and in some cases dramatically improved accuracy for images that deviate from Gaussian shape. Its execution time is typically a few times longer than the CM (corresponding to the number of iterations n require to reach convergence) but 2 to 10 times shorter than the Gaussian mask and 100 to 1000 times shorter than the Gaussian fit even with initial parameters chosen favorably in order to minimize “runaway” unstable cases that do not converge at all.

5. Experimental Results

In order to test the VWCM algorithm in a realistic experimental setting, we tracked the motion of microparticles using digital video microscopy. Our samples consisted of 190 nm diameter fluorescent dye-labeled polystyrene microspheres dispersed and dried (immobilized) on a glass cover slip. The sample was illuminated by a 488 nm solid-state laser, and fluorescence was collected through a water-immersion objective (magnification 60x, *NA* 1.2) and separated from

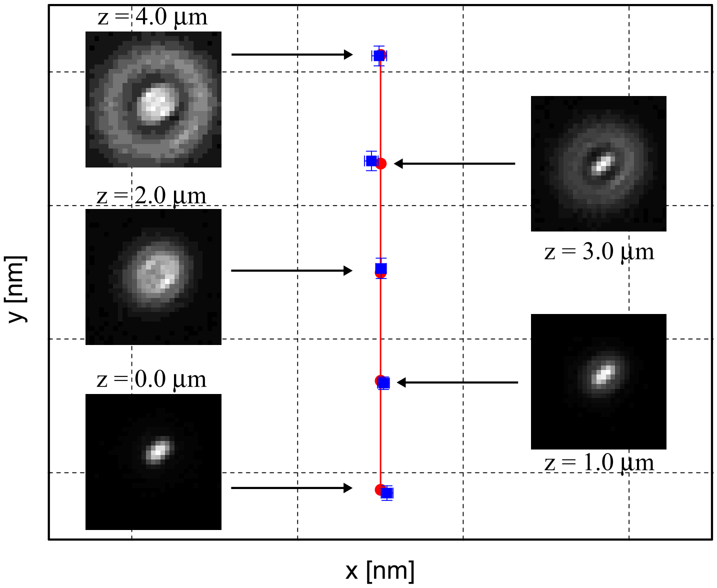

Fig. 4. VWCM position estimates (blue squares) for a binary aggregate of two microparticles stepped in 100 nm increments parallel to the image plane and 1 μm increments out of the focal plane. Approximately 30 images were taken at each position. Error bars represent one standard deviation in the estimate positions, and are inside the data points for some cases. The standard deviation along x ranged between 4.0 nm and 5.6 nm for this series. The red circles are a guide to the eye, spaced at 100 nm intervals.

the excitation by a dichroic filter. Images were obtained with a CCD camera operating at 30 frames per second. Using a three-axis piezoelectric stage, we displaced particles with nanometer precision, and for each image we located the object using the four algorithms discussed in section 4.

In figure 3, we show the distribution of estimated particle positions when a single in-focus particle was scanned in 100 nm increments over a 5 × 5 grid. At each point, approximately 30 images were captured and the resulting estimated position distributions are plotted as circles with 2σ radius (for clarity). This type of data, consisting of in-focus images within a large field of view, represents a "best-case" scenario for the Gaussian fit and mask algorithms (in both accuracy and execution time), since the point-spread function for this case is well-approximated by a Gaussian with constant shape. The data in the figure show that the VWCM eliminates the bias in the CM and gives position estimates commensurate with the Gaussian fit and mask. The latter algorithms are roughly two times more accurate than the VWCM for this in-focus case, but the VWCM executes faster and with no input parameters.

To see how the VWCM performs with images of a more complex nature, we captured images of an asymmetric aggregate of (most likely) two microspheres as it is moved outside the focal plane of the microscope. This asymmetric, extended object was moved in 100 nm increments over five steps while simultaneously defocusing by 1 μm at each step. The resulting images and position estimates are shown in figure 4. Despite the complicated, asymmetric particle shape, the VWCM algorithm tracks the particle motion with high fidelity. For data of this type, the initial parameters for the Gaussian fit and the mask size for the Gaussian mask algorithm would need to be tailored for each image in order to achieve satisfactory tracking. In contrast, the VWCM requires no adjustment for these (or any other) images.

6. Conclusions

In this paper, we described a new algorithm for finding the center of mass of a compact image while suppressing the bias due to a flat background. This VWCM algorithm iteratively centers the particle in a “virtual window” where there is no bias. We demonstrated the VWCM both numerically and experimentally, confirming its predicted simplicity and robustness. It is well-suited for real-time applications in which particles of various sizes and shapes will be tracked simply, robustly and accurately. In the future, we hope to extend our virtual-window method to treat images with a sloped (non-constant) background.

Acknowledgements

The authors gratefully acknowledge Peter Carmichael for stimulating discussions. A. B. is supported by the National Research Council.